Peer-Sensitive ObjectRank
– Valuing Contextual Information in Social Networks

Andrei Damian, Wolfgang Nejdl, and Raluca Paiu

L3S Research Center / University of Hanover
Deutscher Pavillon, Expo Plaza 1, 30539 Hanover, Germany
`{damian,nejdl,paiu}@l3s.de`

Abstract. Building on previous work on how to model contextual information for desktop search and how to implement semantically rich information exchange in social networks, we define a new algorithm, *Peer-Sensitive ObjectRank* for ranking resources on the desktop. The new algorithm takes into account different trust values for each peer, generalizing previous biasing PageRank algorithms. We investigate in detail, how different assumptions about trust distributions influence the ranking of information received from different peers, and which consequences they have with respect to integration of new resources into one peer's initial network of resources. We also investigate how assumptions concerning size and quality of a peer's resource network influence ranking after information exchange, and conclude with directions for further research.

1 Introduction

Due to the boom of web search engines and powerful ranking algorithms like Google PageRank, Web search has become more efficient than PC search. The recent arrival of desktop search applications, which index all data on a PC, promises to increase search efficiency on the desktop. Even with these tools, searching through our personal documents is inferior to searching the documents on the web, though.

The main problem with ranking on the desktop comes from the lack of links between documents, the foundation of current ranking algorithms (in addition to TF/IDF measures). By gathering semantic information from user activities, from the contexts the user works in we build the necessary links between documents. This context information can then be shared between peers in a social network, providing another important source of context information which can be integrated into the local context network of a peer. In this case, however, it is very important to distinguish between items received from different peers of the social group, to take their different background, experience or reliability into account. *Peer-sensitive ObjectRank*, as defined and investigated in this paper, describes how to do just that.

As motivating scenario, let us consider our L3S Research Group context and within this group, Bob, Alice and Tom as three members who exchange information. We assume that Bob mails Alice a document which he sent to the DELOS Workshop. As Bob is one of the authors and therefore has already all the important context for the paper, in this first email, Alice will not only receive the article, but also its immediate context relevant for the research group. From the references included, Alice decides that two of them are of particular interest for her and she sends back an email to Bob requiring additional information about those. As an answer, she receives from him the associated context information, containing the references that Bob has already downloaded. So the context information will be exchanged progressively, from the immediate context to the more distant one. Tom also sends Alice an email, containing a paper on ranking algorithms. From all the papers referenced by it, Tom has read three of them, so that the corresponding context will also include these references.

Bob and Tom play different roles in the L3S research group: Bob is a professor, while Tom is a Ph. D. student. This translates into different levels of trust on Alice's scale, which means that Bob will be higher ranked than Tom, or other authors not known to her. Additionally, the recommendations Alice receives from Bob are more important for her than those received from Tom.

In the next section we provide an overview on how to represent and exchange contextual information based on RDF metadata and specify how to model importance and influence between resources in such a contextual information graph. In Section 3 we discuss how this information is used to rank these resources at a peer and how our ranking algorithm - *Peer-Sensitive ObjectRank* - can take different trust values for the peers participating in the social network into account. In Section 4 we analyse in detail the influence of different trust distributions on the ranking algorithm. We conclude with a short summary and further research issues to be explored.

2 Representing Context and Importance

Motivation. Context information describes all aspects important for a certain situation: ideas, facts, persons, publications, and many more, including all relevant relationships as well as interaction history. Current desktop search prototypes fall short of utilizing any desktop specific information, especially context information, and just use full text index search. In our scenario we clearly need to use additional context information, and specifically want to exploit the *CiteSeer, Browsing* and *Desktop contexts*. A detailed description of these contexts can be found in [3] and [5].

Scenario specific annotation ontologies. For creating the ontologies describing the contexts we are focusing on [3, 5] we use two different shapes: circles and rectangles in order to designate classes and class attributes respectively. We use classes whenever we want to attach importance/rank to entities, attributes otherwise. In addition to the information which resources are included in a specific context, we also want to know how important or valuable these resources are. We therefore have to specify how to express this information, in order to use it for ranking search results.

Authority transfer annotations. Annotation ontologies describe all aspects and relationships among resources which influence the ranking. The identity of the authors, for example, influences our opinion of documents so "author" should be represented explicitly as a class in our publication ontology. We also have to specify how importance of one resource influences the importance of another resource. To do this, we build upon a recent variation of PageRank, ObjectRank [1], which has introduced the notion of authority transfer schema graphs. These graphs extend ontologies by adding weights and edges in order to express how importance propagates among the entities and resources inside the ontology. These weights and edges represent the authority transfer annotations, which extend our context ontologies with the information we need to compute ranks for all instances of the classes defined in the context ontologies.[1]

3 Peer-Sensitive ObjectRank

Our contextual information graphs not only add additional information to the resources on our desktop, they also connect them. This makes it possible to use link-based algorithms like PageRank to enhance ranking in addition to the usual TF/IDF-based methods. In this section we will revisit ObjectRank [1] which can build upon our annotation ontologies to provide PageRank-based ranking for our linked desktop resources and then generalize it to take different trust values for the peers in a social network into account - important for supporting reliable information sharing in such a network.

3.1 PageRank / ObjectRank

Link-based algorithms like PageRank base their computation of rankings on the link structure of the resources, in our case specified by our annotation ontologies and the corresponding metadata. Rank computation is then done using the well-known PageRank formula:

$$r = d \cdot A \cdot r + (1 - d) \cdot e \tag{1}$$

The random jump to an arbitrary resource from the data graph is modelled by the vector *e*. *A* is the adjacency matrix which connects all available instances of the existing context ontology on one's desktop. When instantiating the authority transfer annotation ontology for the resources existing on the users' desktop, the corresponding matrix *A* will have elements which can be either 0, if there is no edge between the corresponding entities in the data graph, or the value of the weight assigned to the edge determined by these entities, in the authority transfer annotation ontology, divided by the number of outgoing links of the same type. According to the formula, a random surfer follows one of the outgoing links of the current page, with the probability specified by *d*, and with a probability of *(1-d)* he jumps to a randomly selected page from the web graph. The *r* vector in the equation stores the ranks of all resources in the data graph, computed iteratively until a certain threshold is reached. [5] presents an example on how this computation is done.

3.2 Peer-Sensitive ObjectRank

Motivation. In our distributed scenario each user has his own contextual network/ context metadata graph and for each node in this network the appropriate ranking computed by the algorithm, as previously described. When sharing resources within the social network, we also exchange the contextual information associated with these resources. This, of course, translates into different ranking values for the items existing on the

---

[1] In contrast to ObjectRank, we do not compute a keyword-specific ranking, but a global one.

desktop. If a user receives resources from several members of his interest group, for which he has different levels of trust, he would like to have higher rank values for the items received from his most trusted neighbours.

Biasing on Peers. The key insight in taking different trust values for different peers into account is that we can represent different trust in peers by corresponding modifications to the e vector. Specifically, we will modify it in a way such that the resources received from highly trusted members have a higher probability to be reached through a random jump than the resources received from less trusted peers. In our formula, this means that the e vector has high values for the positions corresponding to items received from important neighbours, and lower values for the other ones.

In order to be able to create such a "peer-sensitive jumping vector", we have to keep track of the provenance of each resource. We introduce the following notions:

$$originates(r_i, P_n) = \begin{cases} 1 \text{ , if } r_i \text{ is in the original set of resources of peer } P_n \\ 0 \text{ , otherwise} \end{cases}$$

$trust(P_i, P_j) \in [0, 1]$, which represents the trust value of peer P_i for peer P_j.

With these definitions, we can now specify how the e vector is computed:
$e_k(P_i) = max_{j=0}^{N}\{trust(P_i, P_j) \cdot originates(r_k, P_j)\}$,

where N is the total number of peers. This says, that for peer P_i, the probability to jump to the resource r_k is equal to the highest trust value of peer P_i for those peers where this resource originated from. The new formula still guarantees the convergence of our ranking algorithm: it preserves the Markov property of the graph, i.e. each resource can be reached from any other resource. It is a generalization of PageRank biasing, as the e vector now contains more than two different values: basically, for each peer, we have a different value (thus the name *"Peer-Sensitive ObjectRank"*).

Assumptions on Trust Distributions. Based on these definitions, it is obviously important to investigate which influence trust has, if it has any and if so, how much. Besides, which difference does it make for ranking if we consider different trust distributions? We will investigate these issues for different kinds of social networks, and focus on the situation when the peers from a social network have merged all their resources to compare the initial rankings and the rankings after the merge. We will analyze situations where: a) peers have the same trust in one another, b) trust values are evenly distributed (linear distribution) and c) trust is unevenly distributed (powerlaw distribution).

4 Evaluation

4.1 General Experiment Setup & Hypotheses

For the general experiment setup we consider n peers that are part of the same interest group and have different numbers of resources on their desktops. For allocating resources to peers, we build the article set of all the papers published at the Semantic Web related tracks of the following conferences: WWW1999, WWW2000, WWW2001, WWW2002 as well as to ECDL2000 and SWWS2001.

For each peer i, we select an initial set of papers and starting from this set, we consider all their references. For each article we query the CiteSeer database and get additional information (authors, conference, PageRank, references). From all references of a paper we select the top k articles based on their PageRank values and allocate to each peer a percentage of these. We call the percentage of the top ranked articles each peer receives the $accuracy$ of the peer. In addition we allocate to the peers a certain $fill\_in$ percentage from the other articles. We repeat the resource selection mechanism described above until we have collected the number of resources we want to allocate to that peer.

Based on the generated resource sets we compute for each peer i the ACP_i adjacency matrix (AuthorsConferencesPapers). After all resources have been exchanged, all peers will have all articles. The matrix for the union of all resources stored on the peers' desktops is generated appropriately.

Our experiments are intended to investigate the following hypotheses: a) powerlaw trust distribution makes peers "consultancy resistant", i.e. more reluctant to highly value resources received from other peers and b) bigger peers influence smaller peers based on their larger and therefore more influential network of resources. The first hypothesis will be tested mainly through our first experiment, the second hypothesis by the second experiment. An additional hypothesis was that the second effect is stronger than the first one. The next subsections will show whether these hypotheses are true or false and will discuss what we can conclude from the results.

4.2 Experiment I - Consequences of Trust Distributions

Setup. We considered a set of three peers, all being members of the same interest group and having approximately the same number of resources on their desktops: peer 0 has 95 papers, peer 1 has 97 papers and peer 2 has 93 papers and the corresponding authors and conferences. We computed the rankings for all resources initially existing at each peer. The results were sorted in descending order of their rank values. We compared the values for the top 10 best ranked resources (tables 1 and 2).

In our experiment we consider the situation where the peers exchange all the resources they have, so that after a while everybody will have everything. Identical resources coming from different peers appear only once in the resulting data graph, but we keep track of the sources from which each resource originated. In order to be able to see the influence of the trust biased rank computation, we first computed the initial rankings for the merged graph, after all peers have exchanged everything. This rank computation is basically pure ObjectRank computation.

For the next step of our experiment we computed again for each peer the rankings for the whole graph of resources, this time taking different trust values into account. We considered a list of trust values following a linear and a powerlaw distribution respectively. For all computations we assumed that each peer has 100% trust in itself and lower trust values for the other peers.

Results. Some of the results we obtained for this experiment are summarized in tables 1 and 2. Table 1 presents the top 10 highest ranked resources initially computed for peer 0. In the columns labelled "Powerlaw Ranks" and "Linear Ranks", for resources originally appearing at peer 0, we include the ranks we obtained for a rank computation with biasing on a powerlaw trust distribution and a linear trust distribution respectively. The "Merged Ranks" column contains the ranks of the initial resources after peer 0's resources have been merged with all the resources of the other peers. Table 2 has the same structure and presents the results for peer 2.

Discussion. The results we obtain for peer 0 when using a powerlaw trust distribution show a strong resistance against new publications. Peer 0's top 10 initial rankings remain the same with the powerlaw trust distribution and the results indicate that peer 0 integrates not the highest ranked publications from the others, but the ones best connected to his initial network. On the 20th place, peer 0 integrates the resource with the 20th highest rank from peer 1. This is also understandable since peer 0 uses the highest trust value for itself (100%), 25% and 11% trust for peers 1 and 2, respectively. This behaviour is similar in the case of peer 1 and 2.

The linear trust distribution is similar to powerlaw in resistance, though here the rank of others seems to be more important than in the powerlaw case. The 20th highest place in the rank hierarchy is occupied by the resource initially appearing on the 1st place in the top 20 initial rank list of peer 1. Peer 0 assigns 60% trust for peer 1 and 40% trust for peer 2. The same effect can be seen at peers 1 and 2: peer 1 integrates the first ranked resources from peer 0 and peer 2, the resource coming from peer 0 appearing before the resource originating at peer 2. This can be explained not only by the fact that peer 1 assigns 60% trust to peer 0 and 40% to peer 2, but also by the fact that peer 0 has better resources than peer 2, according to the accuracy they use when selecting the references of a paper. For peer 2, the top 20 highest ranked resources include a new resource, which originally appeared after the best 20 ranked resources. This new resource is a conference: its rank has increased since peer 2 received more papers presented at this conference.

Ranks				**Resource**
Powerlaw	**Linear**	**Merged**	**Initial**	
1	1	3	1	ITTALKS...
3	3	7	2	A Framework...
2	2	6	3	Integrating...
9	10	> 10	4	MSL: A Model...
4	4	8	5	Pickling...
5	5	9	6	Evolution of...
8	8	> 10	7	Forward...
7	7	> 10	8	The Case...
6	6	> 10	9	Study On...
10	> 10	> 10	10	Semantic Web..

Table 1. Exp. 1: Rankings for Peer 0

Ranks				**Resource**
Powerlaw	**Linear**	**Merged**	**Initial**	
1	1	1	1	PowerBookmarks:...
2	2	2	2	EDUTELLA:...
3	3	5	3	Logical...
4	4	4	4	Implementation...
5	5	> 10	5	An Improved...
6	6	> 10	6	Automating...
9	10	> 10	7	Squeal:...
7	8	> 10	8	Accessing...
8	9	> 10	9	Simplified...
> 10	> 10	> 10	10	Translating...

Table 2. Exp. 1: Rankings for Peer 2

4.3 Experiment II - Consequences of Initial Resource Distributions

Setup. We considered six peers, all members of the same social network, having different numbers of resources following a powerlaw distribution. Like in the previous experiment, we investigated how trust can influence the rankings of the resources existing on one's desktop, so we run the experiment for linear and powerlaw trust distributions.

Results. For analysing the effects of trust biasing we will present the results only for peer 0 (with 300 papers) and peer 5 (with 5 papers) (tables 3, 4 and 5). Table 3 shows the peer 0's highest top 10 ranked resources for an ObjectRank computation with biasing on a powerlaw trust distribution. We compare these rankings with the ranks we obtain for the rank computation on the merged data graph and with the ones we obtain when biasing on a linear trust distribution. For peer 5 we compare the ranks of the top 10 resources for a powerlaw trust biased rank computation with the merged and initial ranks, and also the “Linear Ranks” with the merged and initial ranks.

Discussion. We observed that peer 0’s initial ranks are identical to the ranks computed after resource merging. Rankings computed with a linear as well as with a powerlaw trust distribution are the same as the initial rankings (Table 3). The reason is that peer 0 had already most of the resources contained in the merged data graph. It is obvious in this case that trust biasing plays no role, peer 0’s rankings remaining unchanged. The situation is totally different for peer 5. This peer has initially only 5 papers, plus a certain number of authors and conferences. Even when computing the new rankings for the merged data graph, its original rankings change dramatically. With a powerlaw trust distribution, peer 5 seems to be resistant to new resources. When biasing on a linear trust distribution, peer 5’s ranks change considerably in comparison to the initial rankings (Table 5). Only a few of the resources still appear in the top 20 new reranked list of resources. Things are different when comparing the new rankings with the initially merged rankings (table 5). With one exception, all resources from the top 20 linear ranks also appear in the top 20 merged rank list, though in a new order.

Ranks			**Resource**
Merged	**Initial**	**Powerlaw**	
1	1	1	PowerBookm...
2	2	2	ITTALKS:...
3	3	3	The Semantic...
4	4	4	EDUTELLA:...
5	5	5	XML-GL:...
6	6	6	Accessing...
7	7	7	Extension of...
8	8	8	Automating...
9	9	9	Integrating...
10	10	10	Keys for...

Table 3. Exp.2: Rankings for Peer 0

Ranks			**Resource**
Merged	**Initial**	**Powerlaw**	
5	1	1	XML-GL:...
> 10	3	2	A Query...
> 10	7	3	SWWS2001...
> 10	6	4	ECDL 2000...
> 10	2	5	Learning...
> 10	> 10	6	SemWeb...
> 10	9	7	Steffen S...
> 10	8	8	Alexander...
> 10	> 10	9	SEBD...
> 10	> 10	10	Dan S...

Table 4. Exp.2: Rankings for Peer 5 (Powerlaw Trust Distribution)

Ranks			**Resource**
Merged	**Initial**	**Linear**	
1	> 10	1	PowerBookm...
2	> 10	2	ITTALKS:...
5	1	3	XML-GL:...
3	> 10	4	The Semantic...
4	> 10	5	EDUTELLA:...
> 10	7	6	SWWS2001...
6	> 10	7	Accessing...
> 10	3	8	A Query...
> 10	6	9	ECDL 2000...
9	> 10	10	Integrating...

Table 5. Exp. 2: Rankings for Peer 5 (linear Trust Distribution)

5 Related Work

The idea of biased ranking has first been explored in topic-sensitive PageRank. [6] builds a topic-oriented PageRank, starting by computing off-line a set of 16 PageRank vectors, biased on each of the 16 main topics of the Open Directory Project. At query time, the vectors are combined based on the topics of the query to form a composite PageRank score for pages matching the query. [2] presents a generalization of this approach.

An interesting search and retrieval system for finding semantic web documents on the web is Swoogle [4]. Their ranking scheme uses weights for the different types of relations between Semantic Web Documents (SWD) to model their probability to be explored. This mainly serves to rank ontologies or instances of ontologies. In our approach we have instances of a fixed ontology and the weights for the links model the users’ preferences.

On the aspect of desktop search, [5] presents another way of making use of the contextual information existing on the users’ desktops, exploring how semantically rich complex recommendation structures, represented as RDF graphs, can be exchanged and shared in a distributed social network. The interest groups are specified with the aid of an extended FOAF vocabulary. The recommendations transport the shared context information as well as ranking information, as described in the annotation ontologies.

The idea of personalized ranking is exploited in [7]. Their personalized search engine built on Google's Web API, redirects the search keywords to Google and retrieves the search results. The system dynamically creates a user profile, by recording the "clickthrough data", which implicitly collects the user's feedback to infer her interests. Based on the user profile and on the result items' semantic meanings in WordNet, the system re-ranks Google search results. The re-ranking is computed differently to the approach described in this paper and is based on the computation of the semantic similarity between result items and user profile.

6 Conclusions and Further Work

This paper presented and analysed a new variant of the PageRank algorithm, called *Peer-Sensitive ObjectRank*, for ranking resources on the desktop, which takes into account different trust values, generalizing previous biasing PageRank algorithms. We performed several experiments and the results confirm the two hypotheses we made in the beginning: powerlaw trust distributions make peers consultancy resistant and bigger peers influence smaller peers.

Several further research questions are worth to be investigated. The first one is how to handle and defend against malicious peers sending bad quality resources. Second, we are interested in investigating the case where one user is member of several interest groups, and exchanges information with members of these groups. Then the user would like to have different trust values for the same peer based on the topic under which a certain resource received from this peer is classified. Finally, we want to investigate in more detail additional aspects which might influence Peer-Sensitive ObjectRank values, such as specific connectivity properties of the context graphs available on the desktop.

References

1. A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, Toronto, Sept. 2004.
2. P. Chirita, W. Nejdl, R. Paiu, and C. Kohlshuetter. Using odp metadata to personalize search. In *SIGIR*, Salvador, Aug. 2005.
3. P. A. Chirita, R. Gavriloaie, S. Ghita, W. Nejdl, and R. Paiu. Activity based metadata for semantic desktop search. In *ESWC*, Heraklion, Greece, May 2005.
4. L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, Washington, DC, Nov. 2004.
5. S. Ghita, W. Nejdl, and R. Paiu. Semantically rich recommendations in social networks for sharing, exchanging and ranking semantic context. In *ISWC*, Galway, Nov. 2005.
6. T. Haveliwala. Topic-sensitive pagerank. In *WWW*, May 2002.
7. G. Huang and W. Liu. An online adaptive method for personalization of search engines. In *WISE*, Australia, Nov. 2004.